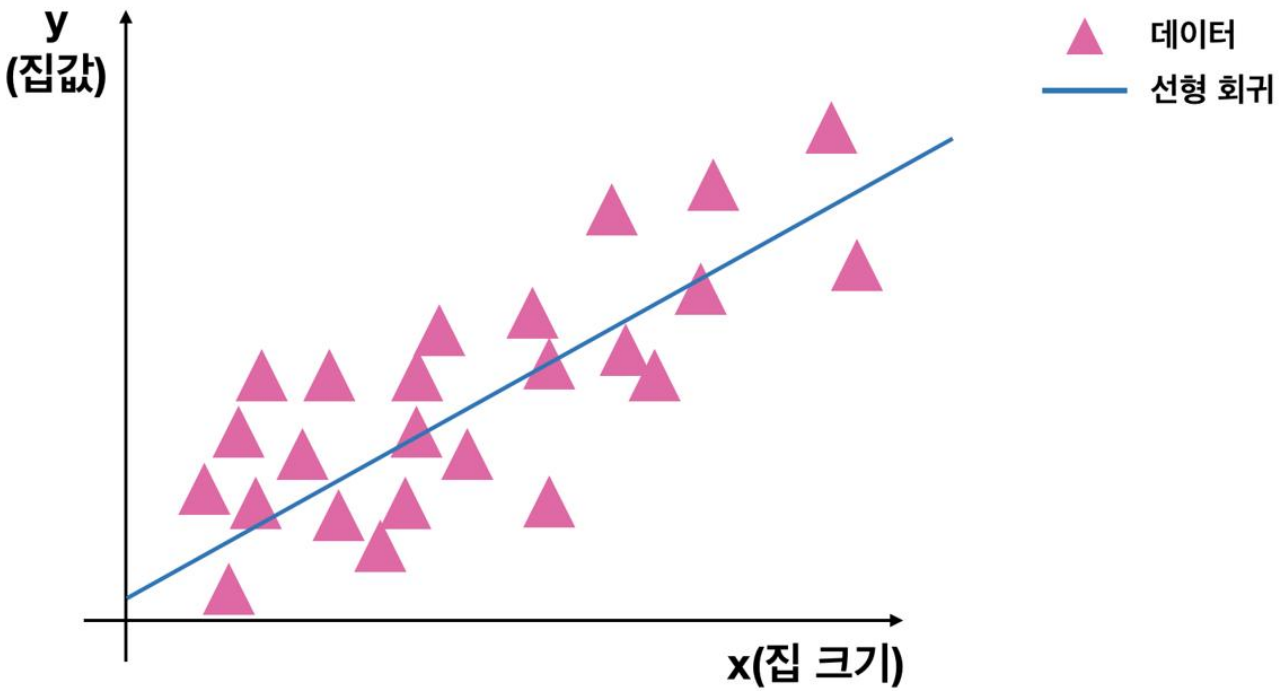
# 05_회귀

D.N.A 1조
심우열 고수영 유소영

# 회귀란?

- 통계의 꽃
- 독립변수와 종속변수의 상관관계를 모델링(예측 모델)
- Y = b1 * X1 + b0 (단순선형회귀)
- 단일회귀 / 다중회귀


y
(집값)
x(집 크기)
데이터
선형 회귀

Classification

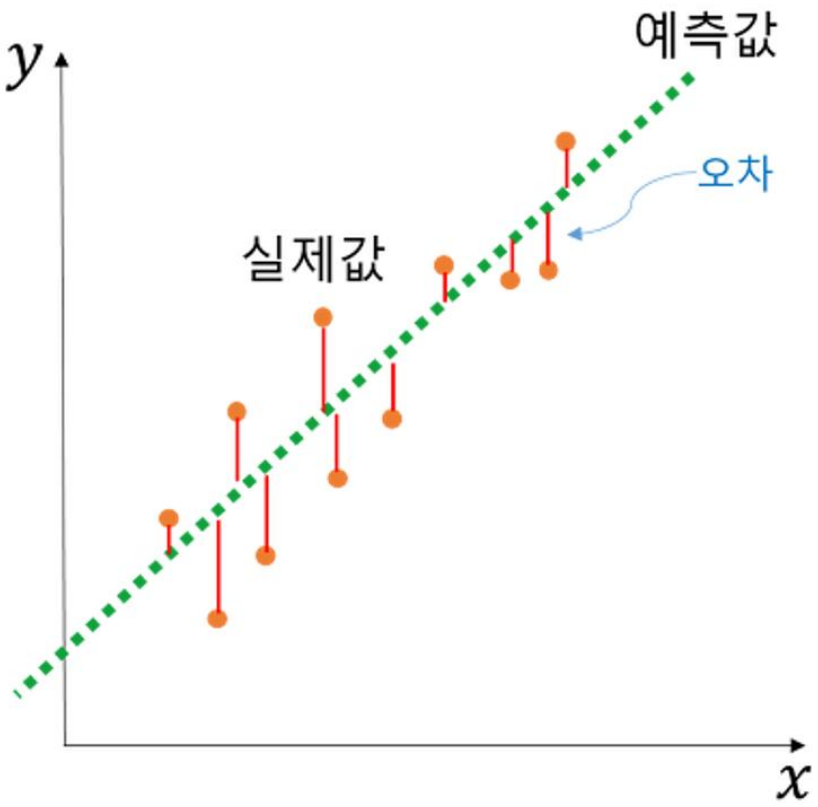
Category 값
(이산값)

Regression

숫자값
(연속값)

# 선형회귀 모델

- 일반 선형 회귀
- 릿지 (Ridge)
- 라쏘(Lasso)
- 엘라스틱넷(ElasticNet)
- 로지스틱 회귀(Logisticc Regression)



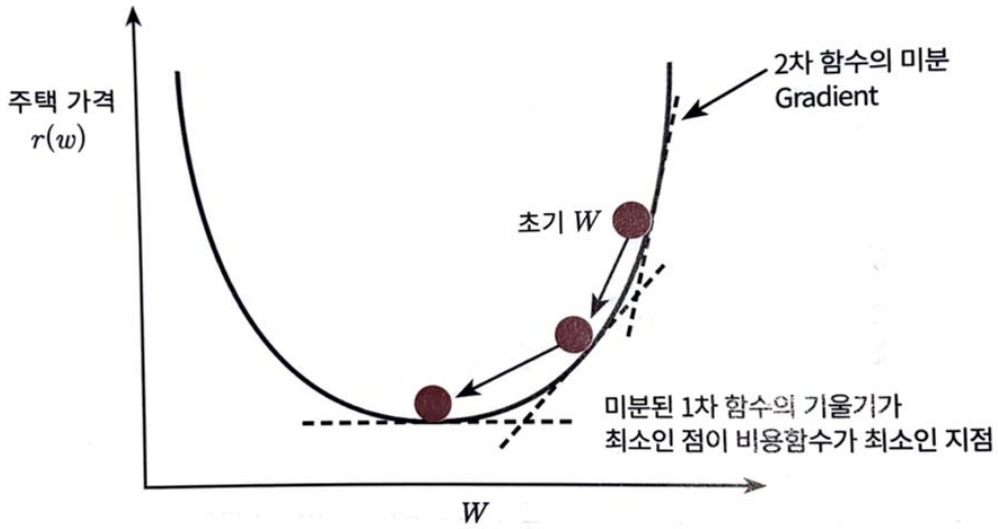

오차를 최소로 만드는게 목표

1. 최소제곱추정

$$: RSS(w_1, w_2) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - (w_0 + w_1 x_i)\right)^2$$

2. 경사하강법

# 경사하강법




Step1
임의의 W값을 설정하고 첫 비용 함수 계산

Step2
W 값을 아래 식으로 업데이트 한 후 비용함수 재계산

$$\frac{\partial R(w)}{\partial w_1} = -\frac{2}{N}\sum_{i=1}^{N} x_i(\text{실제값}_i - \text{예측값}_i)$$

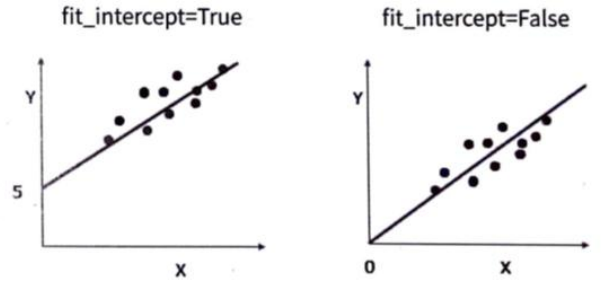
$$\frac{\partial R(w)}{\partial w_2} = -\frac{2}{N}\sum_{i=1}^{N} (\text{실제값}_i - \text{예측값}_i)$$

Stpe3
비용함수 값이 더 이상 감소하지 않을 때 까지 step2반복
그때의 W로 회귀계수 결정

한계 : 모든 학습 데이터에 대해 반복적으로 비용함수 최소화 값 업데이트→ 오래 걸림
극복 : 확률적 경사 하강법 : 일부 데이터만 이용하여 회귀계수 업데이트

# 일반선형회귀

- 규제가 적용되지 않은 선형회귀 모형
- LnearRegression  클래스 이용
- RSS 비용함수를 최소화하는 OLS 추정 방식

| | |
|---|---|
| 입력 파라미터 | **fit_intercept**: 불린 값으로, 디폴트는 True입니다. Intercept(절편) 값을 계산할 것인지 말지를 지정합니다. 만일 False로 지정하면 intercept가 사용되지 않고 0으로 지정됩니다.<br>  <br>**normalize**: 불린 값으로 디폴트는 False입니다. fit_intercept가 False인 경우에는 이 파라미터가 무시됩니다. 만일 True이면 회귀를 수행하기 전에 입력 데이터 세트를 정규화합니다. |
| 속성 | **coef_**: fit() 메서드를 수행했을 때 회귀 계수가 배열 형태로 저장하는 속성. Shape는 (Target 값 개수, 피처 개수).<br>**intercept_**: intercept 값 |

# 회귀 평가 지표

| 평가 지표 | 설명 | 수식 |
|---|---|---|
| MAE | Mean Absolute Error(MAE)이며 실제 값과 예측값의 차이를 절댓값으로 변환해 평균한 것입니다. | $MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert Yi - \hat{Y}i\rvert$ |
| MSE | Mean Squared Error(MSE)이며 실제 값과 예측값의 차이를 제곱해 평균한 것입니다. | $MSE = \frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2$ |
| RMSE | MSE 값은 오류의 제곱을 구하므로 실제 오류 평균보다 더 커지는 특성이 있으므로 MSE에 루트를 씌운 것이 RMSE(Root Mean Squared Error)입니다. | $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2}$ |
| $R^2$ | 분산 기반으로 예측 성능을 평가합니다. 실제 값의 분산 대비 예측값의 분산 비율을 지표로 하며, 1에 가까울수록 예측 정확도가 높습니다. | $R^2 = \frac{\text{예측값 } Variance}{\text{실제값 } Variance}$ |

# 회귀 평가 지표

| 평가 방법 | 사이킷런 평가 지표 API | Scoring 함수 적용 값 |
|---|---|---|
| MAE | metrics.mean_absolute_error | 'neg_mean_absolute_error' |
| MSE | metrics.mean_squared_error | 'neg_mean_squared_error' |
| $R^2$ | metrics.r2_score | 'r2' |

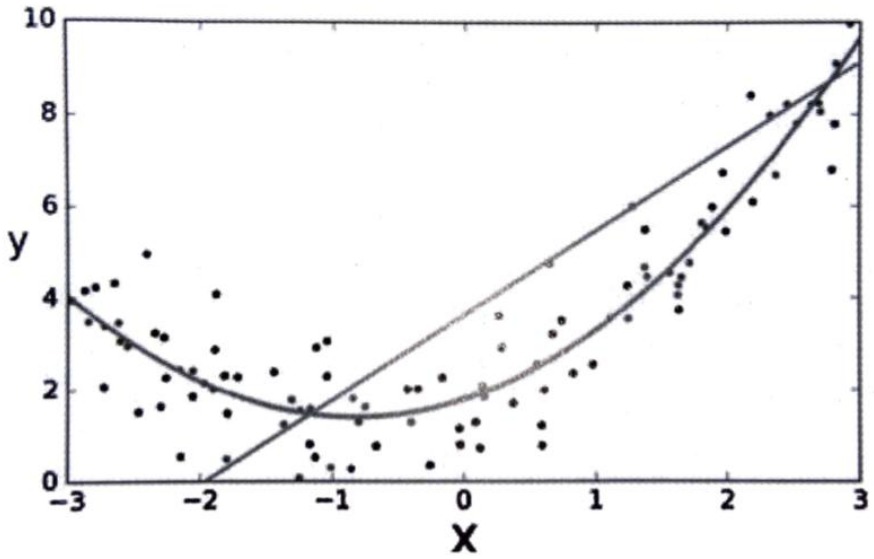
Scoring 함수의 neg_의 의미
사이킷런의 scoring 함수는 그 값이 클수록 좋은 평가로 인식하는데 MSE와 MAE는 작을수록 더 좋은 회귀모형인 모순을 해결하기 위해서 점수를 음수값으로 변경함으로써 그 모순을 해결

# 다항회귀

- 독립변수가 2개 이상인 다항식으로 표현되는 회귀

$$ex)\ y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2$$

- 비선형 함수를 선형 모델에 적용 시키는 방법으로 구현




〈 주어진 데이터 세트에서 다항 회귀가 더 효과적임 〉

# 다항회귀

Step1

데이터 분포에 맞는 차수 결정

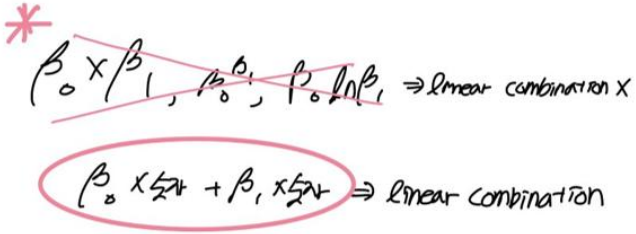
Step2

PolynomialFeatures 클래스를 통해 입력 받은 단항식 피쳐를 다항식 피쳐로 변환

Step3

변환된 피쳐를 통해 LinearRegression 클래스를 통해 다항함수식 유도

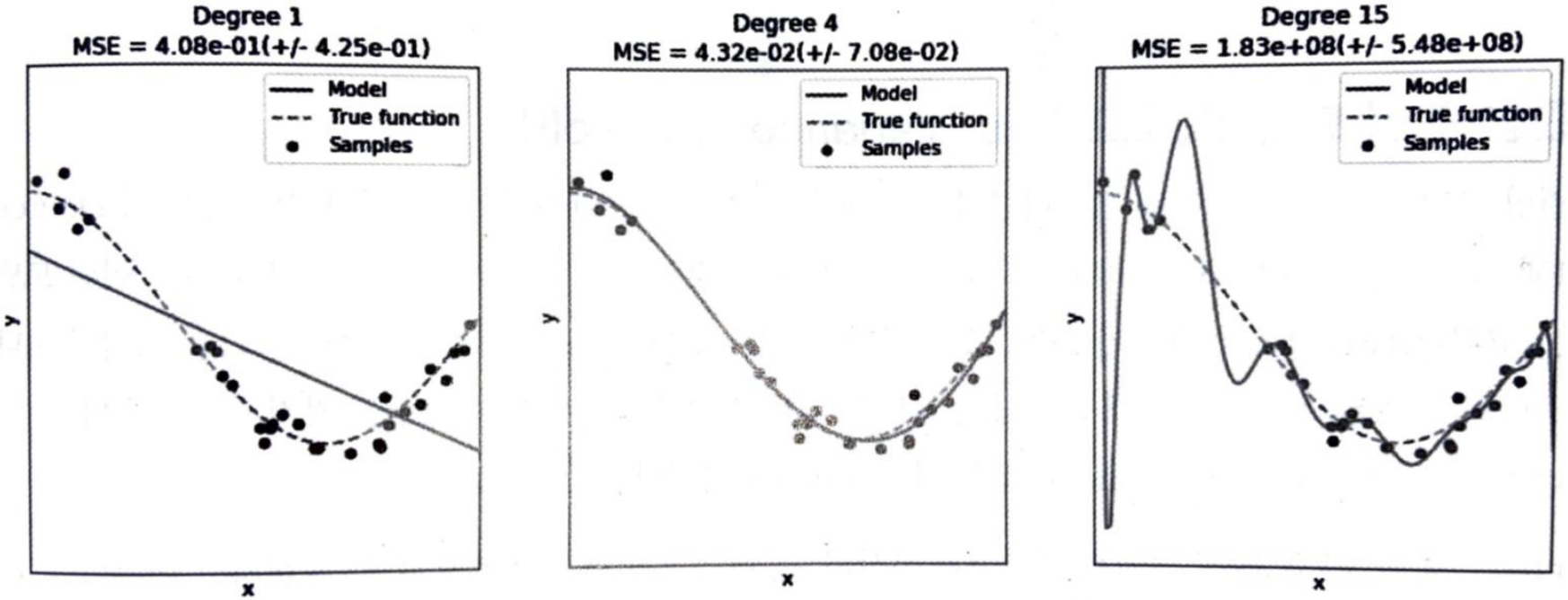
- 다항회귀를 선형회귀라고 하는 이유
  : 선형회귀의 기준이 회귀식의 선형성이 아닌 회귀계수 간의 관계가 선형여부로 갈리기 때문



# 일반선형회귀의 한계점

- 과적합/과소적합

다항회귀 식의 차수가 높아질수록 과적합, 지나치게 낮으면 과소적합 문제 발생



# 일반선형회귀의 한계점

- **편향 분산 트레이드오프**
- 편향과 분산이 모두 낮은 모델이 이상적이나 둘은 트레이드오프 관계로 모두 최소로 낮추는덴 한계가 있음

- 높은편향/낮은 분산 : 과소적합
- 낮은편향/높은분산 : 과대적합
- 편향과 분산이 서로 트레이드오프를 이루면서 오류함수가 최소가 되는 모델을 구축하는 것이 이상적

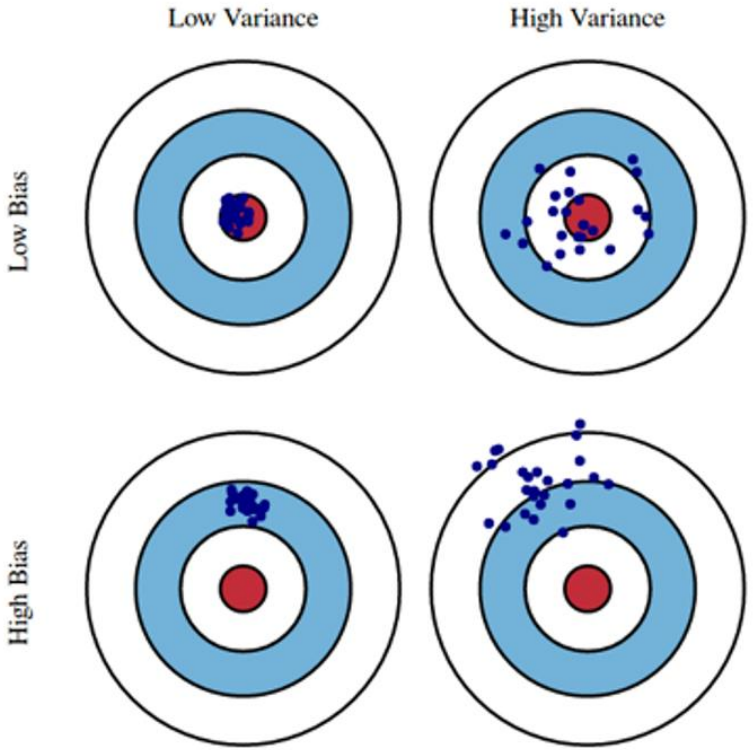



Fig. 1 Graphical illustration of bias and variance.

# 규제선형회귀모델

- 일반선형모델의 과적합 문제 및 지나치게 큰 숫자의 회귀계수 제어 목적
- 규제 정도 척도 : alpha

* 규제선형회귀 비용함수

L1 규제 : $RSS(W) + alpha * ||w||_1$ → 라쏘회귀

L2 규제 : $RSS(W) + alpha * ||w||_2^2$ → 릿지회귀

L1 & L2 : $RSS(W) + alpha1 * ||w||_1 + alpha2 * ||w||_2^2$ → 엘라스틱넷 회귀

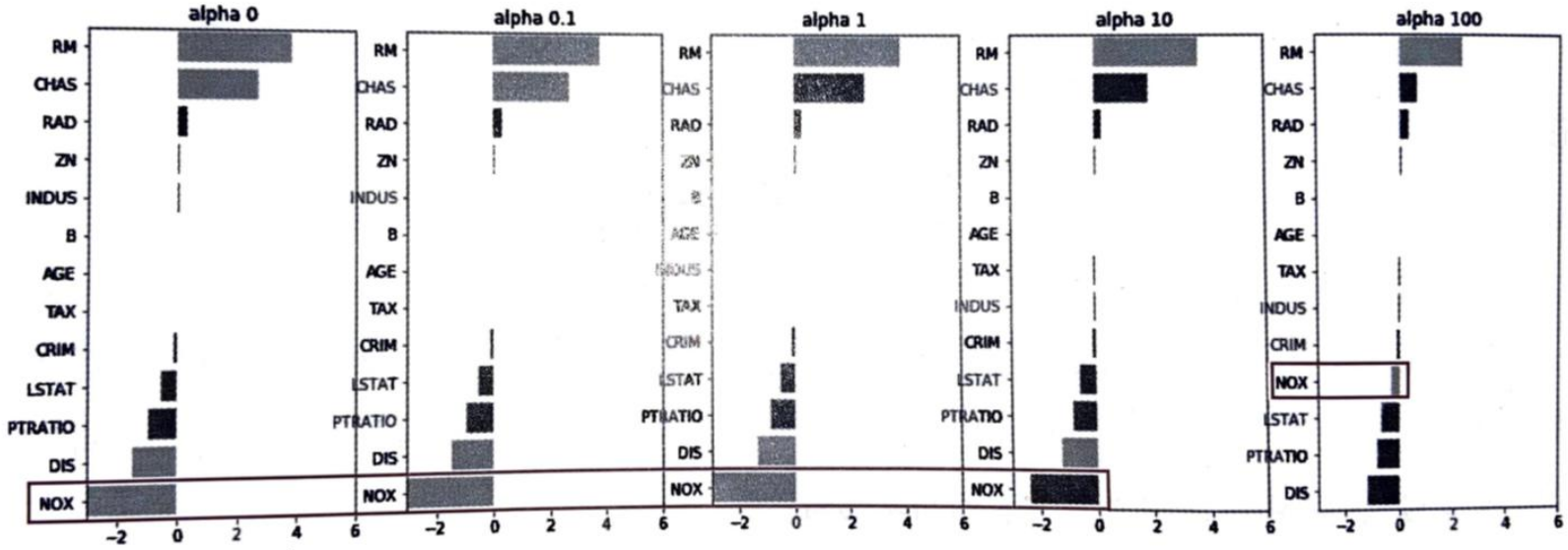
- Alpha 값이 커질 수록 규제 정도가 커지면서 과적합 및 회귀계수 제어정도가 커짐

alpha=0 : 규제가 들어가지 않은 일반선형모델

alpha→무한대 : 규제식이 무한대가 되면서 W를 0에 가깝게 최소화 해야함

# 릿지 회귀(Ridge)

- L2 규제를 통해 크게 결정된 회귀계수를 작게 만드는 규제선형회귀모델
- $RSS(W) + alpha * ||w||_2^2$ 식을 최소화 하는 것이 목적
- 사이킷런의 Ridg 클래스 이용, 주로 파라미터 – alpha



# 라쏘 회귀(Rasso)

- W의 절대값에 제어를 가하는 L1규제를 적용하여 불필요한 회귀계수를 0으로 만들어 없애는 회귀 모델
- $RSS(W) + alpha * ||w||_1$ 식을 최소화 하는 것이 목적
- 사이킷런의 Lasso 클래스 이용, 주요 파라미터 – alpha
- 릿지회귀에 비해 alpha 값에 민감하게 반응

```
#######  Lasso #######
alpha 0.07 일 때 5 folds 의 평균 RMSE : 5.618
alpha 0.1 일 때 5 folds 의 평균 RMSE : 5.621
alpha 0.5 일 때 5 folds 의 평균 RMSE : 5.672
alpha 1 일 때 5 folds 의 평균 RMSE : 5.776
alpha 3 일 때 5 folds 의 평균 RMSE : 6.189
```

| | alpha:0.07 | alpha:0.1 | alpha:0.5 | alpha:1 | alpha:3 |
|---|---|---|---|---|---|
| RM | 3.785460 | 3.698943 | 2.494509 | 0.946786 | 0.000000 |
| CHAS | 1.436287 | 0.957097 | 0.000000 | 0.000000 | 0.000000 |
| RAD | 0.270327 | 0.274112 | 0.277118 | 0.264175 | 0.061867 |
| ZN | 0.049026 | 0.049179 | 0.049528 | 0.049169 | 0.037231 |
| B | 0.010326 | 0.010327 | 0.009532 | 0.008291 | 0.006510 |
| NOX | -0.000000 | -0.000000 | -0.000000 | -0.000000 | 0.000000 |
| AGE | -0.011675 | -0.010006 | 0.003630 | 0.020927 | 0.042495 |
| TAX | -0.014287 | -0.014567 | -0.015440 | -0.015209 | -0.008602 |
| INDUS | -0.041924 | -0.036425 | -0.005109 | -0.000000 | -0.000000 |
| CRIM | -0.097061 | -0.096788 | -0.082662 | -0.063423 | -0.000000 |
| LSTAT | -0.561179 | -0.569509 | -0.656853 | -0.761433 | -0.807679 |
| PTRATIO | -0.765456 | -0.771003 | -0.759070 | -0.723199 | -0.265072 |
| DIS | -1.176150 | -1.160121 | -0.936447 | -0.669009 | -0.000000 |

# 엘라스틱넷 회귀(ElasticNet)

- L1과 L2 규제를 결합한 회귀
- $RSS(W) + alpha1 * ||w||_1 + alpha2 * ||w||_2^2$를 최소화 하는 목적
- 라쏘 회귀가 alpha 값에 민감한 성질로 인하여 회귀계수 값도 급격히 변동한다는 한계를 완화 모델
- 두 규제를 모두 사용함으로써 수행시간이 증가된 단점
- 사이킷런 ElasticNet 클래스 이용, 주로파라미터 – alpha, l1_ratio

  alpha : alpha1+alpha2

  l1_ratio : alpha1/(alpha1+alpha2)

  →l1_ratio=1 : L1규제, l1_ratio=0 : L2규제

라쏘

| | alpha:0.07 | alpha:0.1 | alpha:0.5 | alpha:1 | alpha:3 |
|---|---|---|---|---|---|
| RM | 3.785460 | 3.698943 | 2.494509 | 0.946786 | 0.000000 |
| CHAS | 1.436287 | 0.957097 | 0.000000 | 0.000000 | 0.000000 |
| RAD | 0.270327 | 0.274112 | 0.277118 | 0.264175 | 0.061867 |
| ZN | 0.049026 | 0.049179 | 0.049528 | 0.049169 | 0.037231 |
| B | 0.010326 | 0.010327 | 0.009532 | 0.008291 | 0.006510 |
| NOX | -0.000000 | -0.000000 | -0.000000 | -0.000000 | 0.000000 |
| AGE | -0.011675 | -0.010006 | 0.003630 | 0.020927 | 0.042495 |
| TAX | -0.014287 | -0.014567 | -0.015440 | -0.015209 | -0.008602 |
| INDUS | -0.041924 | -0.036425 | -0.005109 | -0.000000 | -0.000000 |
| CRIM | -0.097061 | -0.096788 | -0.082662 | -0.063423 | -0.000000 |
| LSTAT | -0.561179 | -0.569509 | -0.656853 | -0.761433 | -0.807679 |
| PTRATIO | -0.765456 | -0.771003 | -0.759070 | -0.723199 | -0.265072 |
| DIS | -1.176150 | -1.160121 | -0.936447 | -0.669009 | -0.000000 |

엘라스틱넷

| | alpha:0.07 | alpha:0.1 | alpha:0.5 | alpha:1 | alpha:3 |
|---|---|---|---|---|---|
| RM | 3.570126 | 3.410274 | 1.915894 | 0.937179 | 0.000000 |
| CHAS | 1.332117 | 0.980900 | 0.000000 | 0.000000 | 0.000000 |
| RAD | 0.278304 | 0.282881 | 0.300449 | 0.289167 | 0.147089 |
| ZN | 0.050074 | 0.050586 | 0.052860 | 0.052126 | 0.038281 |
| B | 0.010200 | 0.010145 | 0.009182 | 0.008373 | 0.007029 |
| AGE | -0.010084 | -0.008248 | 0.007777 | 0.020360 | 0.043445 |
| TAX | -0.014519 | -0.014810 | -0.016044 | -0.016213 | -0.011417 |
| INDUS | -0.044641 | -0.042520 | -0.023093 | -0.000000 | -0.000000 |
| CRIM | -0.098392 | -0.098179 | -0.068550 | -0.073471 | -0.019596 |
| NOX | -0.178164 | -0.000000 | -0.000000 | -0.000000 | -0.000000 |
| LSTAT | -0.575540 | -0.588404 | -0.694327 | -0.760714 | -0.800276 |
| PTRATIO | -0.779878 | -0.785066 | -0.791232 | -0.738850 | -0.423093 |
| DIS | -1.189080 | -1.173268 | -0.975771 | -0.725297 | -0.031389 |

동일한 alpha 값에 대해서 라쏘 회귀에 비해 엘라스틱넷 회귀에서 0이 된 회귀 계수가 더 적음

# 선형회귀모델 데이터변환

- 타깃값이나 피처값이 왜곡된 분포를 띌 경우 예측성능에 부정적 영향을 미칠 수 있음
- 이를 방지하기 위해 회귀모델을 적용하기 전 데이터에 대해 스케일링/정규화 작업 수행
- 피처데이터셋 스케일링/정규화 방법

1. StandardScaler 클래스를 이용해 평균이 0, 분산이 1인 표준 정규 분포를 가진 데이터 세트로 변환하거나 MinMaxScaler 클래스를 이용해 최솟값이 0이고 최댓값이 1인 값으로 정규화를 수행합니다.
2. 스케일링/정규화를 수행한 데이터 세트에 다시 다항 특성을 적용하여 변환하는 방법입니다. 보통 1번 방법을 통해 예측 성능에 향상이 없을 경우 이와 같은 방법을 적용합니다.
3. 원래 값에 log 함수를 적용하면 보다 정규 분포에 가까운 형태로 값이 분포됩니다. 이러한 변환을 로그 변환(Log Transformation)이라고 부릅니다. 로그 변환은 매우 유용한 변환이며, 실제로 선형 회귀에서는 앞에서 소개한 1, 2번 방법보다 로그 변환이 훨씬 많이 사용되는 변환 방법입니다. 왜냐하면 1번 방법의 경우 예측 성능 향상을 크게 기대하기 어려운 경우가 많으며 2번 방법의 경우 피처의 개수가 매우 많을 경우에는 다항 변환으로 생성되는 피처의 개수가 기하급수로 늘어나서 과적합의 이슈가 발생할 수 있기 때문입니다.

- 타깃값이 왜곡됐을 경우 로그변환을 일반적으로 수행. 단, 변환 후 원복 복구가 어려움

# 선형회귀모델 데이터변환

| 변환 유형 | alpha값 | | | |
|---|---|---|---|---|
| | alpha=0.1 | alpha=1 | alpha=10 | alpha=100 |
| 원본 데이터 | 5.796 | 5.659 | 5.524 | 5.332 |
| 표준 정규 분포 | 5.834 | 5.810 | 5.643 | 5.424 |
| 표준 정규 분포 + 2차 다항식 | 8.776 | 6.849 | 5.487 | **4.631** |
| 최솟값/최댓값 정규화 | 5.770 | 5.468 | 5.755 | 7.635 |
| 최솟값/최댓값 정규화 + 2차 다항식 | 5.294 | **4.320** | 5.186 | 6.538 |
| 로그 변환 | **4.772** | **4.676** | **4.835** | 6.244 |

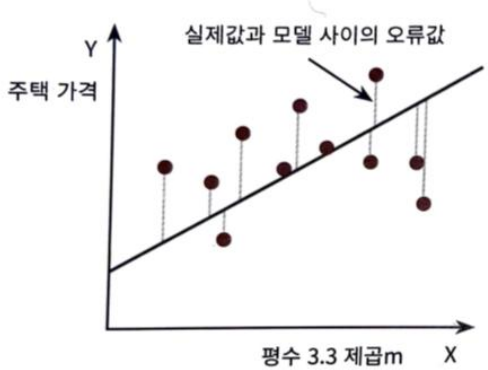
릿지 회귀 모델 적합 시 데이터 변환 유형 별 alpha 값에 따른 RMSE 값 비교

# 로지스틱회귀

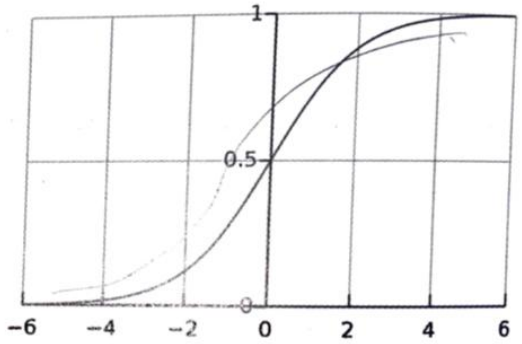
- 선형회귀 방식을 분류에 적용한 알고리즘
- 독립변수에 대한 선형회귀가 아닌 가중치 변수에 대한 선형회귀
- 시그모이드 함수 이용
- 사이킷런의 LogisticRegression 클래스 이용

시그모이드 함수?

$y = \frac{1}{1+e^{-x}}$로 정의되는 함수로, x 값에 상관없이 y의 값은 항상 (0,1)구간 값을 가지는 함수

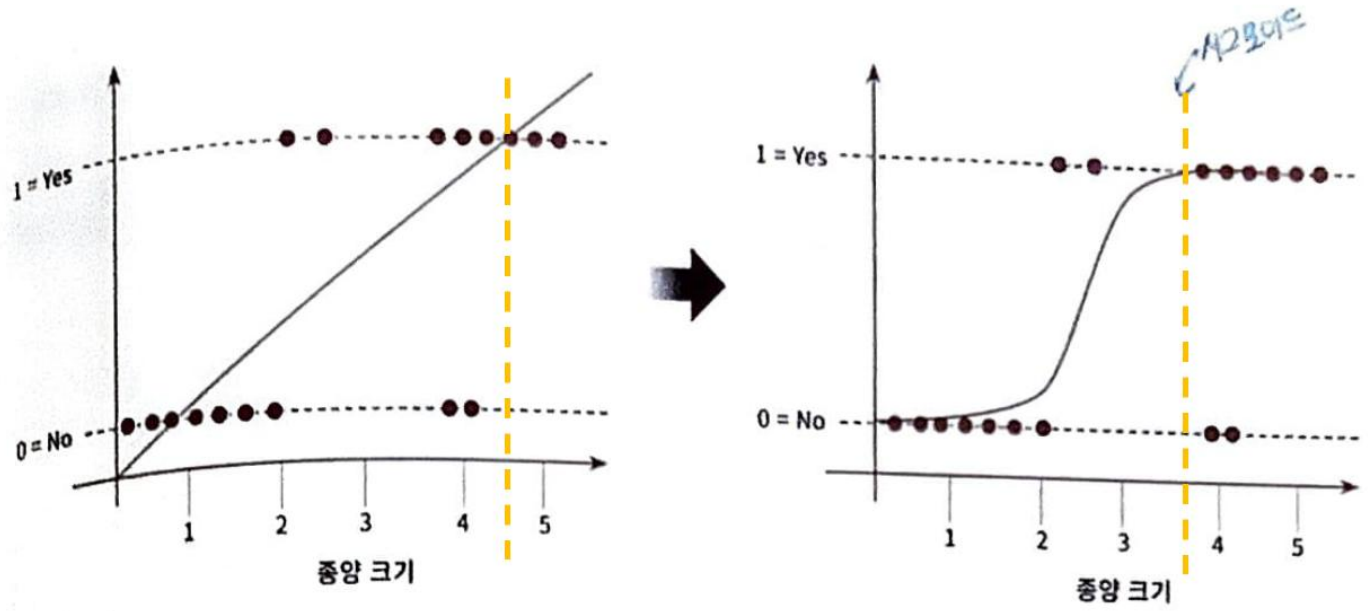


〈 선형 회귀의 선형 함수 〉



〈 로지스틱 회귀의 시그모이드 함수 〉

# 로지스틱회귀

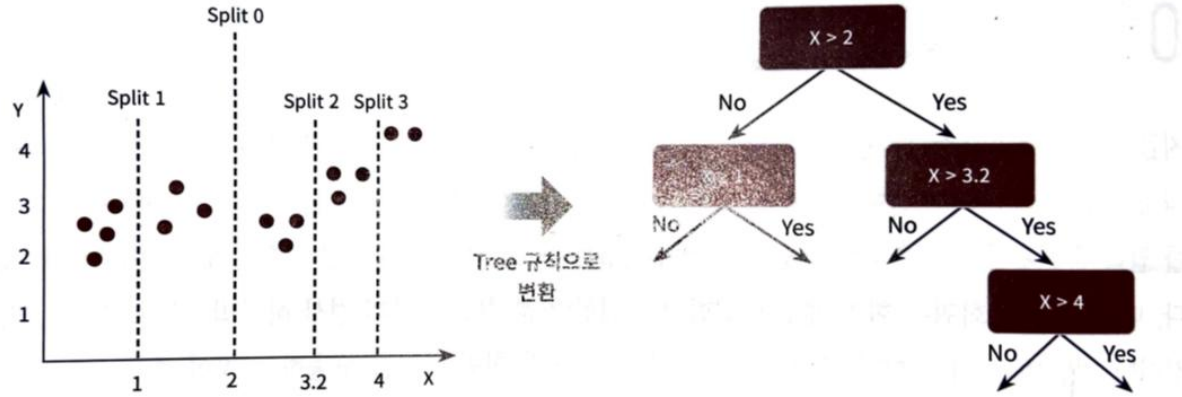


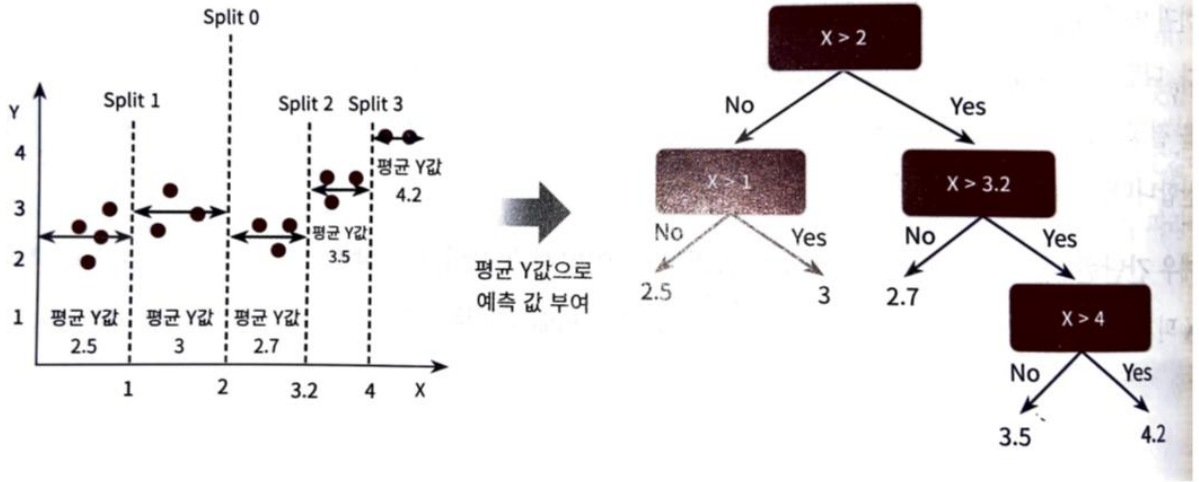
# 회귀 트리

- 회귀를 위한 트리를 생성하고 이에 기반하여 예측하는 비선형회귀모델
- 리프 노드에서 최종 예측 결정 값을 만드는 과정에만 차이 존재(뻗어 나가는 과정을 동일)

  분류 트리 : 특정 클래스 레이블 결정

  회귀 트리 : 리프 노드에 속한 데이터 값의 평균으로 예측값 계산
- 사이킷런의 DecisionTreeResgression 클래스 이용
- 입력파라미터는 분류트리 파라미터와 거의 동일
- 일반 선형회귀 클래스들처럼 coef_를 통해 회귀계수를 알려주는 대신 feature_importances_를 통해 피처별 중요도 파악 가능


Split 0
Split 1
Split 2
Split 3
Y
4
3
2
1
1
2
3.2
4
X
Tree 규칙으로
변환
X > 2
No
Yes
X > 1
X > 3.2
No
Yes
No
Yes
X > 4
No
Yes


Split 0
Split 1
Split 2
Split 3
Y
4
3
2
1
평균 Y값
4.2
평균 Y값
3.5
평균 Y값
2.5
평균 Y값
3
평균 Y값
2.7
1
2
3.2
4
X
평균 Y값으로
예측 값 부여
X > 2
No
Yes
X > 1
X > 3.2
No
Yes
No
Yes
2.5
3
2.7
X > 4
No
Yes
3.5
4.2